AD-A050 497 NORTHROP CORP HAWTHORNE CALIF ELECTRONICS DIV F/G 17/7
AN/BRN-7 COMPUTER PROGRAM SPECIFICATION. VOLUME IX. EXECUTIVE S--ETC(U)
OCT 73 N00039-73-C-0209

UNCLASSIFIED NORT-73-48-VOL-9 NL

| OF |

AD
A050497

END
DATE
FILMED
3 -78
DDC

n Electronics Division NORT

VOL IX

NORTHROP

Electronics Division

- [ ] This submittal applies to AN/BRN-7 (Submarine Ω) only.
- [ ] This submittal applies to AN/SRN-( ) (Hydrofoil Ω) only.
- [x] This submittal applies to both AN/BRN-7 and AN/SRN-( ).

CONTRACT NO: N00039-73-C-0209

PROGRAM NAME: AN/BRN-7

CDRL No: A01D, A01E, A01F

Title of CDRL: Computer Program Design Specification
Computer Subprogram Design Document
Data Base Design Document

Title of Doc: AN/BRN-7 Computer Program Specification
NORT 73-48
Volume 2 thru 13

Date: 1/16/74

Initial Submittal:	X	Release	APS 1-16-74
Resubmittal:		Authentication	

ACCESSION for		
NTIS	White Section	X
DDC	Buff Section	
UNANNOUNCED		
JUSTIFICATION Per DDC Form 50 on file		
BY		
DISTRIBUTION/AVAILABILITY CODES		
Dist.	AVAIL. and/or SPECIAL	
A		

Prepared by: System Integration Unit
Orgn. Name

A4734 X574
Orgn. No. and Ext.

Approved by:

Norman P. Clark — Supervisor of Preparing Orgn.

Bill Tyler — Program Mgr.

E. V. Oman — Proj. Engr./Proj. Mgr.

FORM 6512 (2-72)

NORTHROP

Electronics Division

Cop 0

AD-A050497

NORT 73-48

AN/BRN-7 COMPUTER PROGRAM SPECIFICATION

Volume IX
EXECUTIVE SUBPROGRAM DESIGN

October 12, 1973

Approved by

G. W. Hauser
Director, Engineering
Navigation Department

Volume IX
of the

AN/BRN-7 OMEGA COMPUTER
PROGRAM SPECIFICATION

CONTENTS

SECTION 1

SCOPE

1.1 IDENTIFICATION

Volume I, Submarine OMEGA Computer Program Performance Specification, defines the functional requirements for the Submarine OMEGA Computer Program which is used by the AN/BRN-7 OMEGA Navigation Set. The Navigation set and the OMEGA program together comprise the Submarine OMEGA Navigation System. The tape which defines the computer program is entitled AN/BRN-7 Navigation Program.

Volume II, Submarine OMEGA Computer Program Design Specification, allocates the functional requirements of Volume I to the computer routine and subprogram level. This volume describes the subprogram designated as Executive which has the abbreviation EX in the program listing (Volume XII).

1.2 EXECUTIVE SUBPROGRAM TASKS

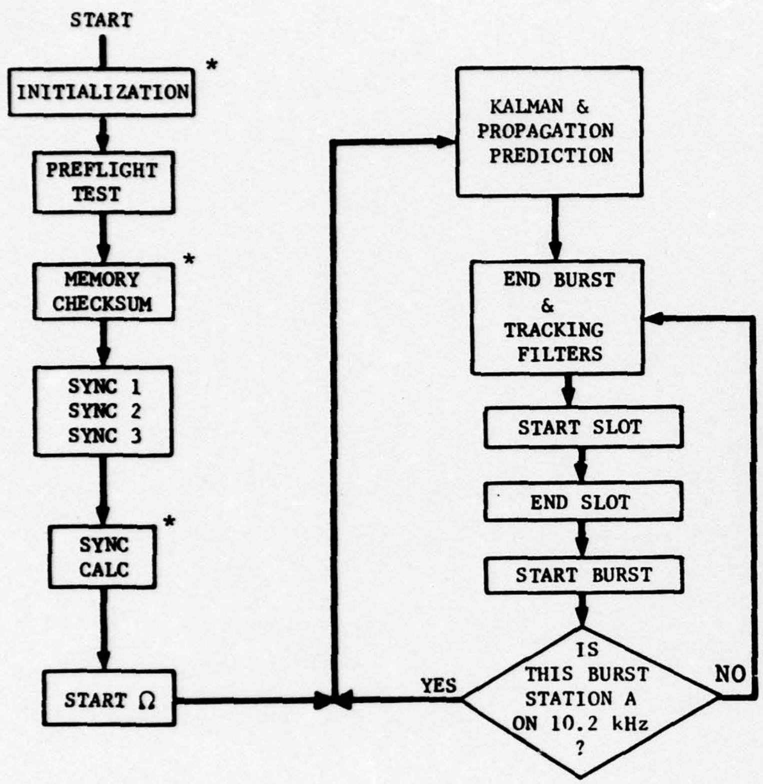
1.2.1 Main Program

Historically, the label Main Program has been given to that primary software routine that encompasses the main body of program calculation and requires a large amount of execution time. The interrupt routine is one that temporarily preempts execution of the Main Program in order to initiate a subroutine of time dependent nature. Here, however, the time dependent and sequential nature of the Submarine OMEGA software has boosted the interrupt routine to eminence and reduced Main Program to the state of marking time, or idling. Main Program will initiate the executive routine at start-up and then will continually check core until interrupted. This is illustrated in Figure 1.

1.2.2 Five MS Interrupt Routine

All timing and sequencing of the OMEGA Program is done by the use of the 5-millisecond interrupt originating from the Precision Frequency Generator in the receiver section of the system.

Every 5 milliseconds the computer program will halt processing of the current software task. It will then check a list of priorities to determine if an Ω-task (high priority task, dependent upon some aspect of incoming signals) should be initiated, and if so then that Ω-task will be initiated. If it is not true that an Ω-task should be initiated then a non-Ω-task (low priority task, not directly dependent on OMEGA burst pattern) may be initiated before returning to the originally interrupted task. In this way the software will always initiate sequencing of those routines which are dependent on the OMEGA burst and slot pattern at their scheduled initiation time.



This main sequence will be interrupted every 0.1 second to perform:

VELOCITY PROCESSING
RIJ UPDATE
AVIONIC I-O TEST

Routines above plus START Ω constitute non-Ω-tasks. All other non-asterisked functions are Ω-tasks.

Note 1: Memory checksum test is performed when no other routine in process.

Note 2: Production Inspection and Production Control Inspection Tests are not shown

*Controlled by Main Program; all others controlled by Five Millisecond Interrupt Routine.

10803

FIGURE 1 INTRODUCTORY FLOW DIAGRAM OF ROUTINES

SECTION 2

APPLICABLE DOCUMENTS

a) Submarine OMEGA Computer Program Performance Specification (Volume I of the Submarine OMEGA Computer Program Specification)

Applicable Sections

3.1 Introduction

3.2 Functional Description

3.3 Detailed Functional Requirements

3.3.2 Signal Input Timing

3.3.16 Built-In Test Programs

b) Submarine OMEGA Computer Program Design Specification (Volume II of the Submarine OMEGA Computer Program Specification).

c) NORT 71-41, NDC-1070 MACRO ASSEMBLER, MAY 1971

d) NORT 68-115A, Detailed Description of NDC-1070 Computer Instructions, Revision A, February 1970.

e) NORT 69-87A, NDC-1070 Flow Chart Program, User's Manual.

SECTION 3

REQUIREMENTS

In order to understand the program description contained in the following pages, it is necessary that the reader will have become familiar with the associated functional requirements found in Volume I, Performance Specification, and with the subprogram allocation found in Volume II, Design Specification.

3.1 DETAILED DESCRIPTION

3.1.1 Reference Labels to Flow Diagrams

The code used to reference the particular block in the flow diagrams, Section 3.2, is as follows: The first number preceded by a p, is the page number found in the upper right corner of the diagrams. This will be followed by a slash sign (/) to separate the page number from the block designator. The designator will either be a mnemonic label (e.g., TEST SYNC), a local label indicated by a dollar sign ($), or an integer. The two types of labels reference the particular information block, on the given page, to which the label is attached. The integer number, n, means that the referenced block is the n^{th} block from the top of the page; p 8/3 would refer to page 8 and the third information designator down.

The label P1/$2+3 refers to page 1, and the 3rd information block after the label $2. P2/7,8,9 refers to Page 2 and the 7th, 8th and 9th blocks.

3.1.2 Flow Diagram Description

3.1.2.1 Main Program

An overview of Main Program is given as Figure 2.

P1/BEGIN, 2,3,4. After initializing the pushdown stacks the program sequences to memory protect. However, this is a function dependent upon the Program Monitor Unit and is not required. The same is true of the OMEGA Debug Subroutine which is a software tool used in conjunction with the PMU permitting operator to read or write data from or into memory. Here, OMEGA Debug operates as a return function.

P1/5. The addition of computer time to previous GMT is an operator convenience in the case of momentary power fluctuations which cause the system to turn off. If the span of power off was short enough the operator will elect to accept the GMT presented by the system display. Following this the memory cleared.

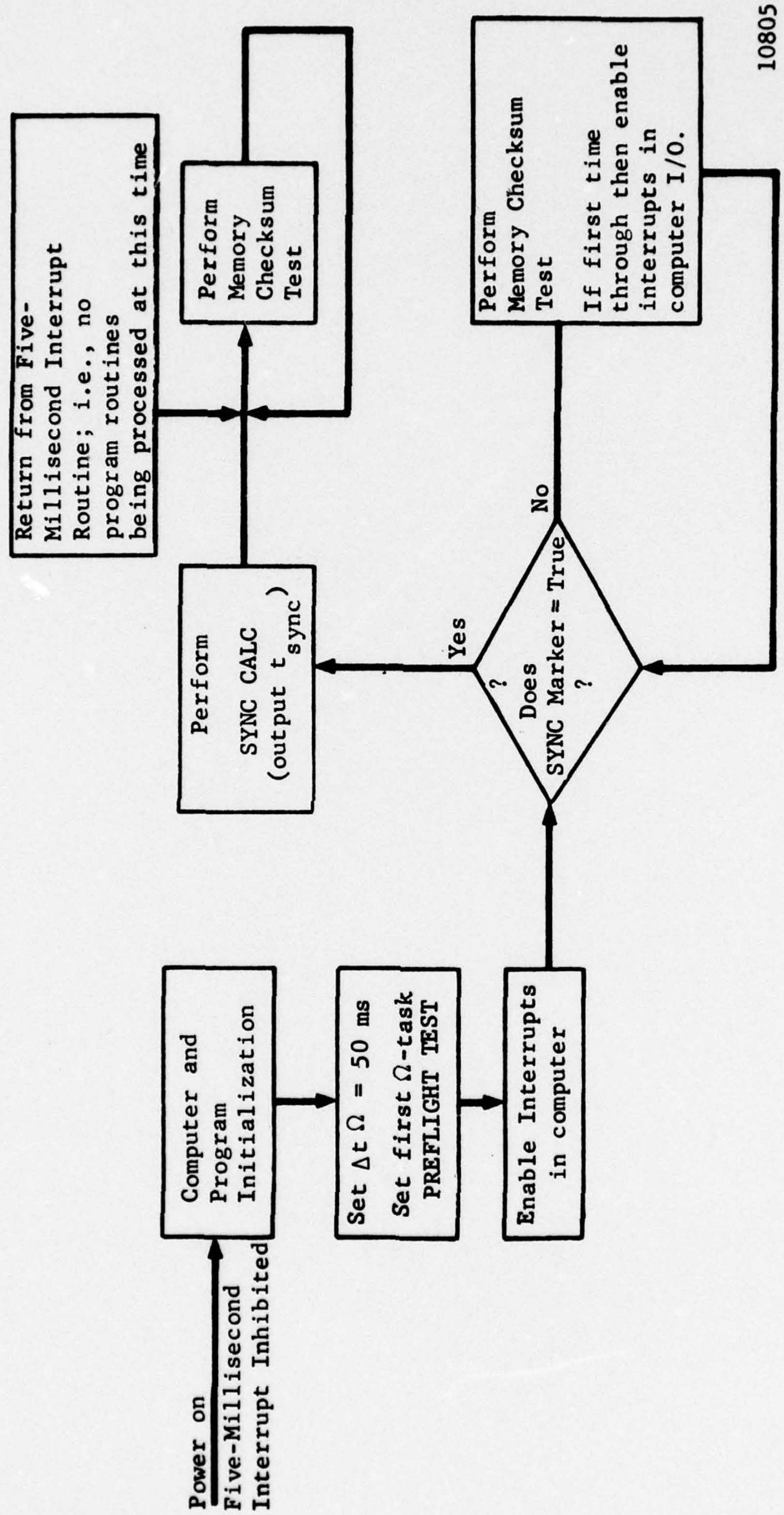


FIGURE 2 FLOW DIAGRAM OF MAIN PROGRAM

P2/1. The Control-Indicator is initialized. The switch mask is a register where a bit is posted when a button/indicator has been established (and illuminated) by the Control-Indicator Procedures as a valid button for depression by the operator in order to continue or initiate procedures. The switch mask is used as a validity check after a button has been depressed. The operation here only enables the upper 16 bits of the switch mask permitting operation of the buttons indicated in this block. The program ID consisting of date of assembly and modification number is displayed on the Control-Indicator Panel.

P2/2. Set $\Delta t\,\Omega$ = .05 second and first Ω-task as Preflight Test. However, the Interrupt routine is at present inhibited and will not be completely enabled until Main Program completes one sequence through Memory Checksum. Consequently, Preflight Test will not occur until 50 milliseconds after completion of the interrupt enable. For description of Preflight Test refer to description of P4/3.

P2/3. Included as an integral part of the Precision Frequency Generator (PFG) is a Digital Phase Comparator which will detect an out-of-synchronization condition of the frequency divider network of the PFG. A BITE signal is generated by the computer program when an out-of-synch condition is detected. The computer will sense this condition and generate a re-synch signal to the PFG. This same signal is generated now to initialize the PFG.

P2/4. The computer has been designed to allow the special I/O to communicate directly with the computer memory. This capability exists only with a preselected block of memory words (see Table 3.2-2, Volume I) and includes both input and output.

The significance of the data, the timing of the transfer and the amount of data are entirely dependent on the design of the special I/O. It is only used for communication with the Submarine OMEGA System and not with support equipment associated with the computer.

The transfer of data to and from the memory can be inhibited by setting the DMA inhibit flip-flop with the LGC instruction. This flip-flop is true when the program start signal is given.

P2/5. The interrupts generated by the special I/O and the support equipment may be inhibited by setting the interrupt inhibit flip-flop with an LGC instruction. This flip-flop is also set true by a program start signal. If an LGC instruction that changes the state of the inhibit flip-flop is being executed at the time an interrupt request is being received, the state of the flip-flop at the completion of the execution of the instruction controls the interrupt. If more than one interrupt request is received by the computer at one time, the highest priority interrupt is the power off interrupt (which cannot be inhibited) followed by the special I/O and then the support equipment interrupts.

P2/6. DMA word 23_{16} controls the selection of system tests plus other miscellaneous signals (see Volume I, Section 3.2.3.3 (h)). Bit 9 controls the Receiver-Computer NO-GO signal which will be posted as zero if the program detects a system problem that can be isolated to the receiver-computer. This will turn on a lamp on the receiver-computer box that must be manually reset. As long as this lamp is on, the SYSTEM MALF Lamp on the Control-Indicator panel will be illuminated.

Bit 10 is the System Malfunction control which will be posted as zero at any time a system malfunction is detected. The system lamp on the C&I panel will also be illuminated when this signal is given. In addition, the SYSTEM MALF lamp will be illuminated if the hardware detects a failure in the basic computer clock or if the Program Monitor signal is not given every 80 ms.

Bits 9 and 10 are now set to one.

The program must execute a SET 000C at least every 80 milliseconds or the system malfunction lamp on the C & I Panel will be turned on. This will assure that a malfunction will be indicated if the program fails to execute the SET instruction due to either a programming error or a computer failure.

P2/8,9; (P3/1,2), The Memory Checksum Test is provided as a test to verify the contents of computer memory. This test sums all of the permanently stored (non-variable) contents of memory. The computer compares the sum against a predetermined value; failure to compare constitutes a failure of the test.

This test is mechanized entirely with the computer software. At the time of program assembly, the assembler program generates the "checksum" for that program assembly and places this value in a storage location. When the program is loaded into the Omega computer, this checksum shall be constant for the program as loaded. If memory is altered in any way from the loaded program, the checksum constant is no longer valid and the test will indicate failure.

Once through Memory Checksum the Interrupt routine is enabled and 50 milliseconds later, Preflight Test will occur. Main Program will remain in this loop until the SYNC marker is set true, whereupon Main Program will perform the SYNC CALC routine. (See Volume III, Synchronization Subprogram).

P3/1,2. Henceforth Main Program will perform the Memory Checksum Test when the 5 MS Interrupt routine is not performing an Ω-task or a non-Ω-task.

3.1.2.2 Executive

This is the Five-Millisecond Interrupt routine and is shown in Figure 3.

P4/1. The TIME counter is incremented by one bit. The TIME counter counts 5 millisecond interrupts since computer turn-on.

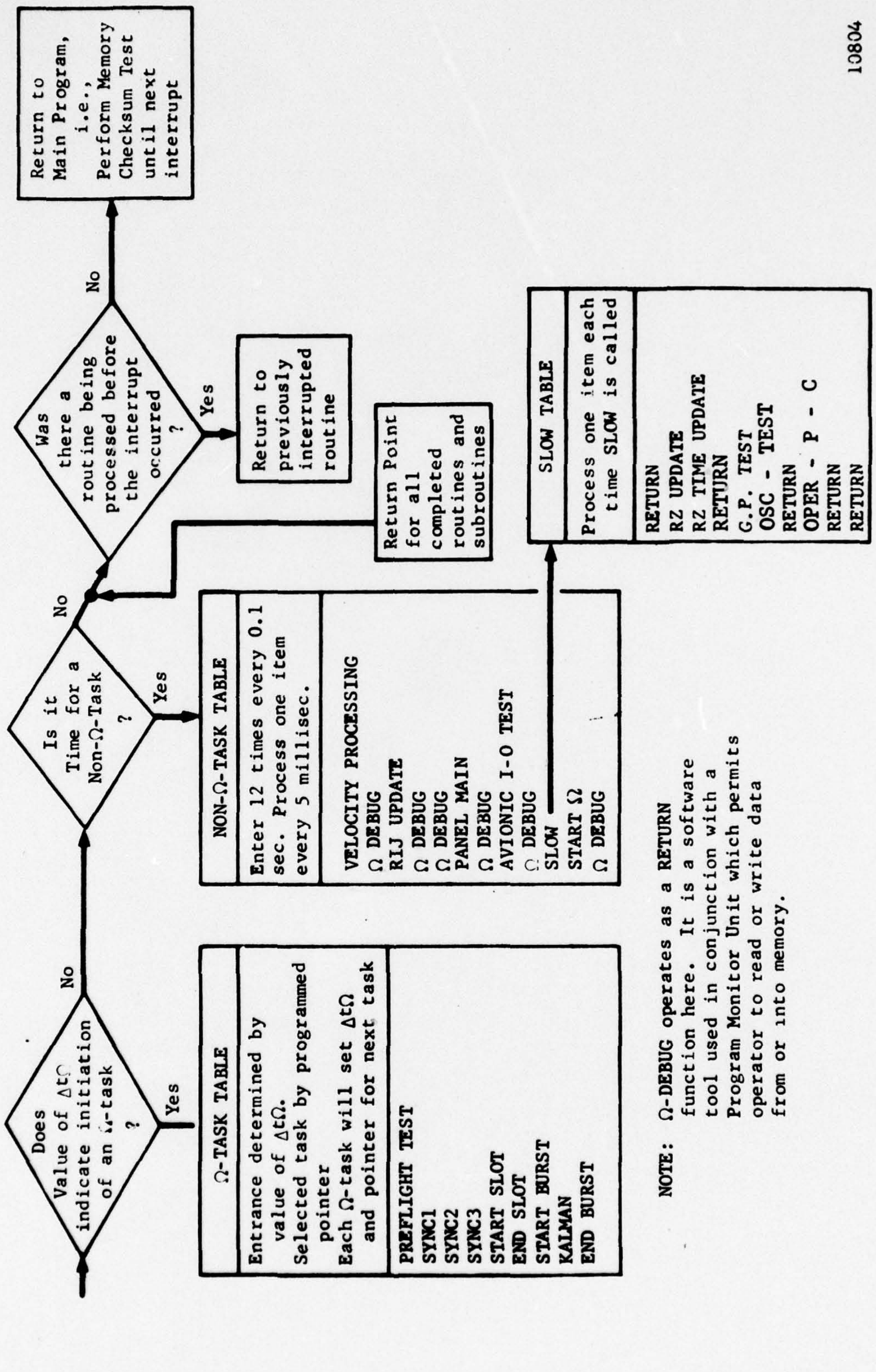
P4/2. See description of P2/6.

P4/3. The term $\Delta t\Omega$ determines that interrupt at which an Ω-task will be initiated. At this time the program will branch to that Ω-task indicated (which has been selected by the previous Ω-task). One of the first instructions of the Ω-task will be to select the next Ω-task and the time of execution, $\Delta t\Omega$. A summary of the Ω-tasks follows (See Figure 3).

a) Preflight Test (Volume XI): Sets $\Delta t\Omega$ = 0.2 second
next Ω-task = SYNC1
Performs the following tests (found in Volumes I and XI)

- Receiver
- Phase to Digital
- Phase Counter
- DMA
- GP Self Test
- Memory Checksum
- Avionic I/O
- Omega Pre-amp Test
- Oscillator Test

b) SYNC1 (Volumes II, III): Sets $\Delta t\Omega$ = 0.15 second
next Ω-task = SYNC2
Sets up Antenna Switching Matrix.



NOTE: Ω-DEBUG operates as a RETURN function here. It is a software tool used in conjunction with a Program Monitor Unit which permits operator to read or write data from or into memory.

FIGURE 3 FLOW DIAGRAM OF THE FIVE-MILLISECOND INTERRUPT ROUTINE

c) SYNC 2 (Volumes II, III): Sets $\Delta t_\Omega = 0.1$ second
next Ω-task as SYNC 3
initialization of time and storage areas for synchronization.

d) SYNC 3 (Volumes II, III): Sets $\Delta t_\Omega = 0.1$ second
next Ω-task as SYNC 3
Performs data collection. Upon completion of synchronization a non-Ω-task, START OMEGA, is executed which sets $\Delta t_\Omega = 0.2$ seconds and next Ω-task as Kalman (see description of START OMEGA, flow diagram page 7)

e) KALMAN (Volume VI): Sets $\Delta t_\Omega = 0.005$ second
next Ω-task as END BURST
Synthesizes tracking filter data with phase velocity data from Propagation Prediction, and provides corrections to position, velocity and OMEGA oscillator parameters.

f) END BURST (Volume IV): Sets $\Delta t_\Omega = 0.15$ second
next Ω-task as START SLOT
Selects the proper antenna and calibrate signals for the slot measurement and computes the phase and phase variance of the station bursts.

g) START SLOT (Volume IV): Sets $\Delta t_\Omega = 0.1$ second
next Ω-task as END SLOT
initializes input registers for slot data collection.

h) END SLOT (Volume IV): Sets $\Delta t_\Omega = 0.15$ second
next Ω-task as START BURST
Antenna select for station bursts. Computes scale factors and makes noise and phantom measurements.

i) START BURST (Volume IV): Sets $\Delta t_\Omega = 0.695$, 0.795, 0.895 or 0.995 second
next Ω-task as KALMAN.
Increments the station count, sets the time of burst and performs base station selection.

P4/5,6

If 0.1 second or more has elapsed since the last non-omega task then execute the next non-Ω-task. Otherwise continue and return to the subprogram which was interruped to process this 5-millisecond interrupt.

P5/1,2,3,4

The table pointer keeps track of which item in the non-Ω-task table should be initiated. The 5-millisecond interrupt routine will cause halts in processing; sometimes to initiate the higher priority Ω-tasks; at other times reentry is immediate. If a non-Ω-task is unfinished, reentry is by the interleaved program sequencing (see Volume II) via the push down stack.

T5/NON-OMEGA TASK TABLE

a) Velocity Processing is the first table entry executed.

This algorithm processes the velocity and heading data which is input from either the E.M. Ship's Log (velocity) and the Mark 19 Repeater (heading), and/or the respective manual inputs from the Control-Indicator. If the data are from the external sources, the inputs will be smoothed before processing. This is not necessary for the manual inputs.

b) Omega Debug

Described under P1/BEGIN

c) RIJ Update

The iterative updating of the position matrix is performed by converting the velocity over the last iteration into the effective angular rotation about each of the system axes. These rotations are then used to generate the rotation update matrix. The update matrix is multiplied by the previous position matrix to yield the change of position, and this result is then added to the old position. Periodically the Kalman routine will be generating positional corrections. These will be added to and processed with the rotations generated from the velocity.

d) Panel Main

Display procedures selected by the operator and necessitating update are processed. The Panel Main task has a 10 counter so that Panel Main procedures are updated once per second.

e) Analog-to-Digital I/O Test

A conversion, analog-to-digital, is made by the Avionic I/O using the 400 Hz synchro reference signal as a reference. The resultant digital value is verified by the computer program.

This test is executed every 0.1 second during system operation. It examines the A-to-D test word and indicates a failure when it is out of limits an excessive number of times. The test is divided up into sets

of 12 samples. A failure consists of 3 or more bad readings in 5 consecutive sets.

f) SLOW

Discussed under page 6 description.

g) Start Omega

The purpose of the routine is to bridge the gap between the end of the synchronization process and the start of OMEGA navigation, and to synchronize the initiation of the non-Ω-task table with that of the Ω-task table. The objectives can be stated as follows:

1) Set Kalman as the next Ω-task and calculate Δt_Ω.
This insures that Kalman will be initiated on the lowest processing level.

2) Set $t_{NON-\Omega} = \Delta t_\Omega - 0.02$

This insures that R_{ij} update (position) is accomplished 5 milliseconds before Kalman processing. The purpose here is to make available to Kalman the most recent position data.

3) Set START Ω marker = false.

This insures that the START Ω routine will not be processed during normal program iterations.

See description of flow diagram page 7, START OMEGA.

P6/SLOW

This routine will cycle through a table of 10 entries executing one task each time, thereby yielding routines executed once per second. Five of the entries are return functions. The other five entries are as follows (equations in Volume X, Control-Indicator Subprogram Design):

- *RZ Update*

  This routine iterates the angular velocity of the moving destination (second submarine) then calculates its present position. The routine depends upon operator insertion of second submarine initial position, velocity and heading.

- RZ Time Update

  Uses the current position of second submarine (calculated in RZ Update) and own position, heading and speed to calculate the Point of Intercept between the two vehicles.

- OSC - TEST

  The oscilator drift test is performed. This test is designed to detect out of limits values of the oscilator.

- GP Test 9 or Computer Logic Test)

  General Description: The computer Logic Test is designed to verify that the Arithmetic and Control section of the computer hardware is functioning in a normal manner.

  The test consists of execution of the basic instructions of the computer being executed and being checked for proper bit patterns in appropriate registers. Upon completion of this a "sample problem" is executed, primarily using the ARCTAN routine of the normal program storage. This routine was selected because it uses about 90% of the instruction repertoire of the computer.

- Programmer Controller Display

  This program will display go - no go conditions on the Programmer Controller when it is connected.

P7/START OMEGA

As already described under description P5/5,6, the purpose of this routine is to bridge the gap between the end of the synchronization routine and the operational OMEGA navigation program.

From synchronization the time associated with the rise time on station A, 10.2, is related to the computer clock time by t_{sync}. The Combinational Filter (Kalman) will be initiated 0.775 second later; and it is desirable to begin velocity processing and R_{ij} Update as close as possible to the initiation of Kalman processing. Hence, the non-Ω-tasks will be initiated 0.02 second prior to Kalman start. This is also described in Figure 4.

P7/1,2

Upon entry to this routine the START OMEGA Marker, which is set by the synchronization routine upon completion, is read. This is set once and reset at the end of this routine. Henceforth, START OMEGA will immediately return to last interrupted process.

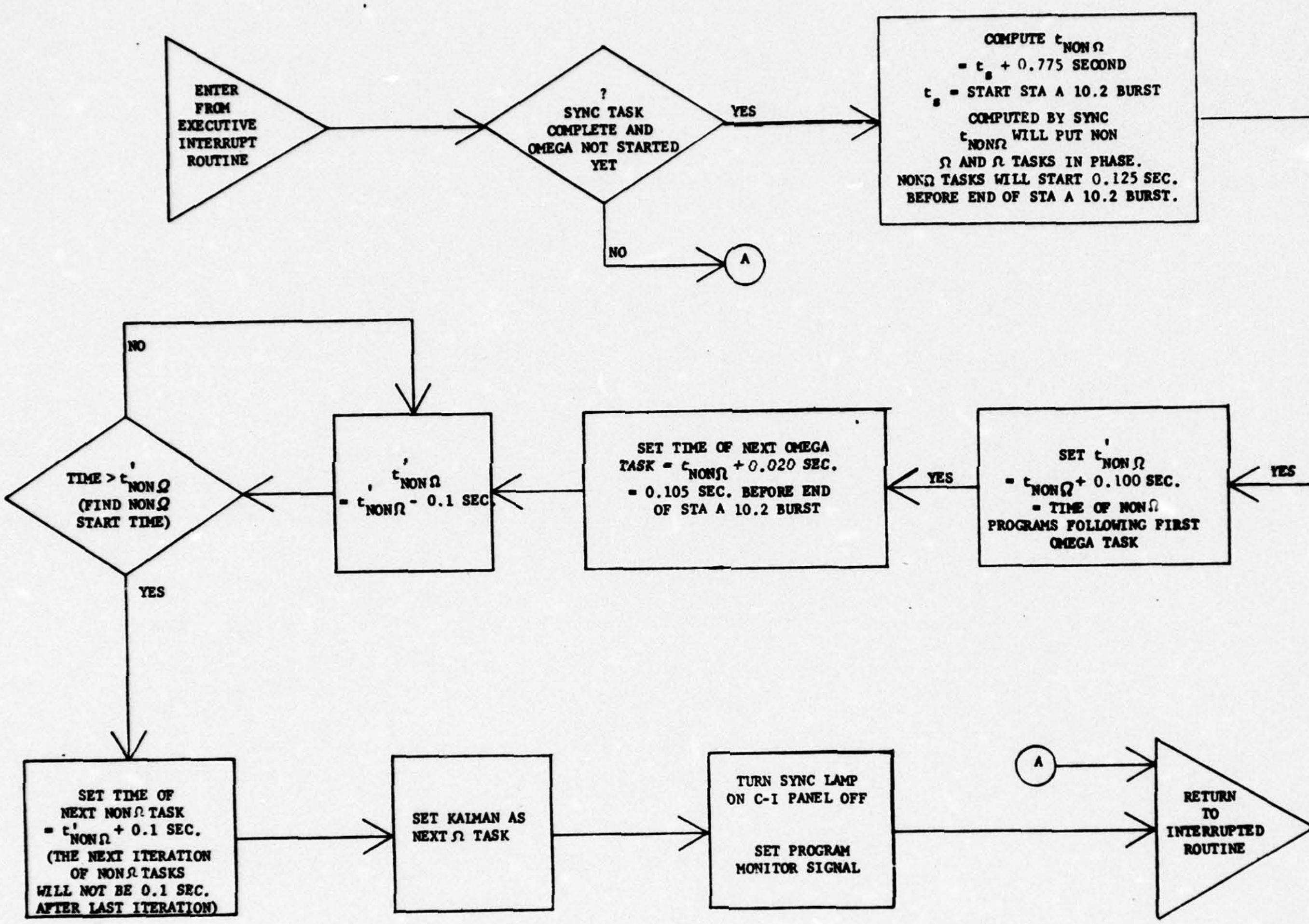


FIGURE 4 START OMEGA (A NON-OMEGA TASK)

P7/3,4

Optimum $t_{NON-\Omega} = t_{sync} + .775$

P7/$1,$1 + 1

The optimum $t_{NON-\Omega}$ is the time at which it is considered optimal for the initiation of non-Ω-tasks. It is the span between the computer start time and the immediately succeeding station A rise time (on 10.2 kHz) plus 0.775 second. Thus, a number of 10-second intervals must be added to put the optimum $t_{NON-\Omega}$ in the future. This is done by comparing n(optimum $t_{NON-\Omega}$) with current time and iterating.

Set $t_{non-\Omega}$ = n (optimum $t_{non-\Omega}$) at exit.

P8/1

Sets $t_{\Omega\text{-task}} = t_{non-\Omega} + 0.02$ and stores

P8/2

Sets $t_{non-\Omega} = t_{\Omega\text{-task}} + 0.08$

P8/3,4

Since $t_{non-\Omega}$ may be seconds in the future the value of 0.1 second is successively decremented until the program can begin processing.

P8/5,6,7,8

Stores time, sets next Ω-task as Kalman, turns SYNC lamp off on Control-Indicator Panel, resets START OMEGA Marker, and conducts Program Monitor Test (see description P2/7).

3.1.2.3 Subroutines and Interrupts

P10/OMEGA TASK PIN through P13/3

This is the subroutine used by each Ω-task to incorporate Δt_Ω into the program timing and the Ω-task into the program sequencing.

Following the omega task subroutine are the list of interrupts and the power off sequence.

3.2 FLOW DIAGRAMS


```
*           MAIN PROGRAM
* THIS ROUTINE IS THE LOWEST LEVEL ROUTINE
* IN THE SYSTEM.  IT IS ENTERED BY THE POWER
* ON INTERRUPT OR WHENEVER A DECISION IS
* MADE TO RESTART.  THIS ROUTINE PERFORMS
* INITIALIZATION, ENABLES DMA AND INTERRUPTS,
* ENTERS THE SYNC COMPUTATIONAL ROUTINE AND
* DOES CHECKSUM.  IT ONLY OPERATES WHEN THERE
* ARE NO OMEGA OR NON OMEGA TASKS REMAINING.
```

BEGIN

INITIALIZE R14 AND R15 TO POINT
TO THE TOP OF THE R14 AND R15
PUSHDOWN STACKS

TURN ON PROTECT IN CASE PROGRAM
MONITOR IS AVAILABLE

SET   ENABLE MEMORY PROTEC
T

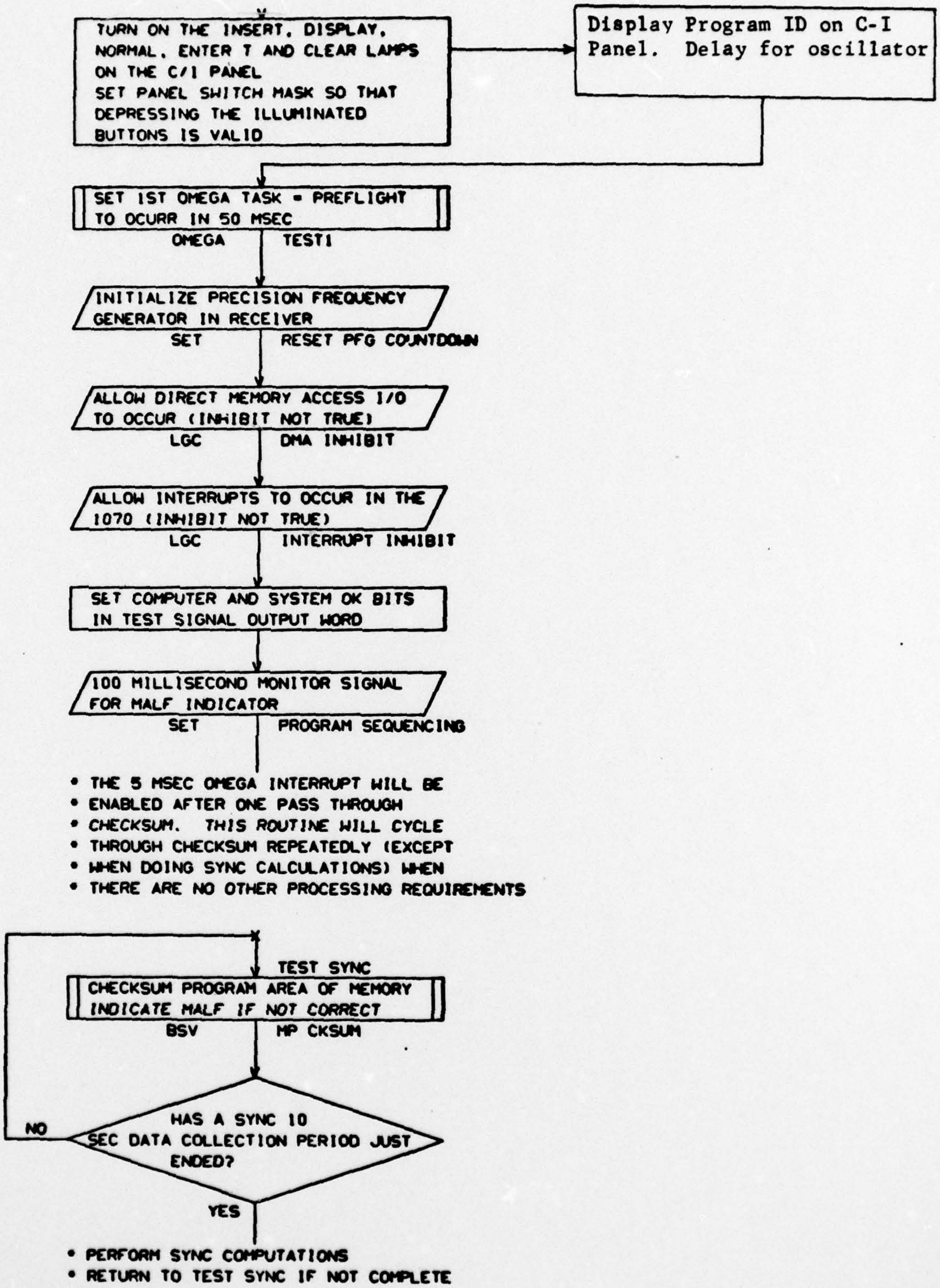
SET UP TO ENTER DEBUG IN SUCH A
WAY THAT IT WILL READ THE INPUT
WORD ON THE FIRST ENTRY WITHOUT
WAITING FOR A BUTTON TO BE
DEPRESSED. THIS WILL ALLOW THE
OPERATOR TO 'HOLD' THE PROGRAM
FROM THE PMU AT TURN ON.

DEBUG - AUTOMATIC RETURN IF NO
PROGRAM MONITOR UNIT AVAILABLE

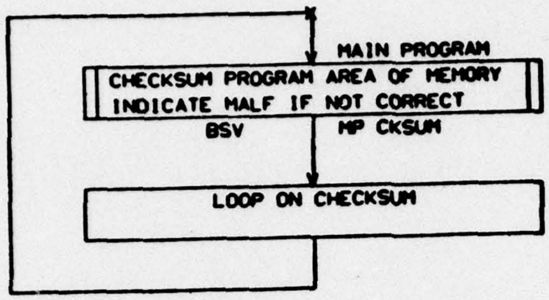
BSV   OMEGA DEBUGS2

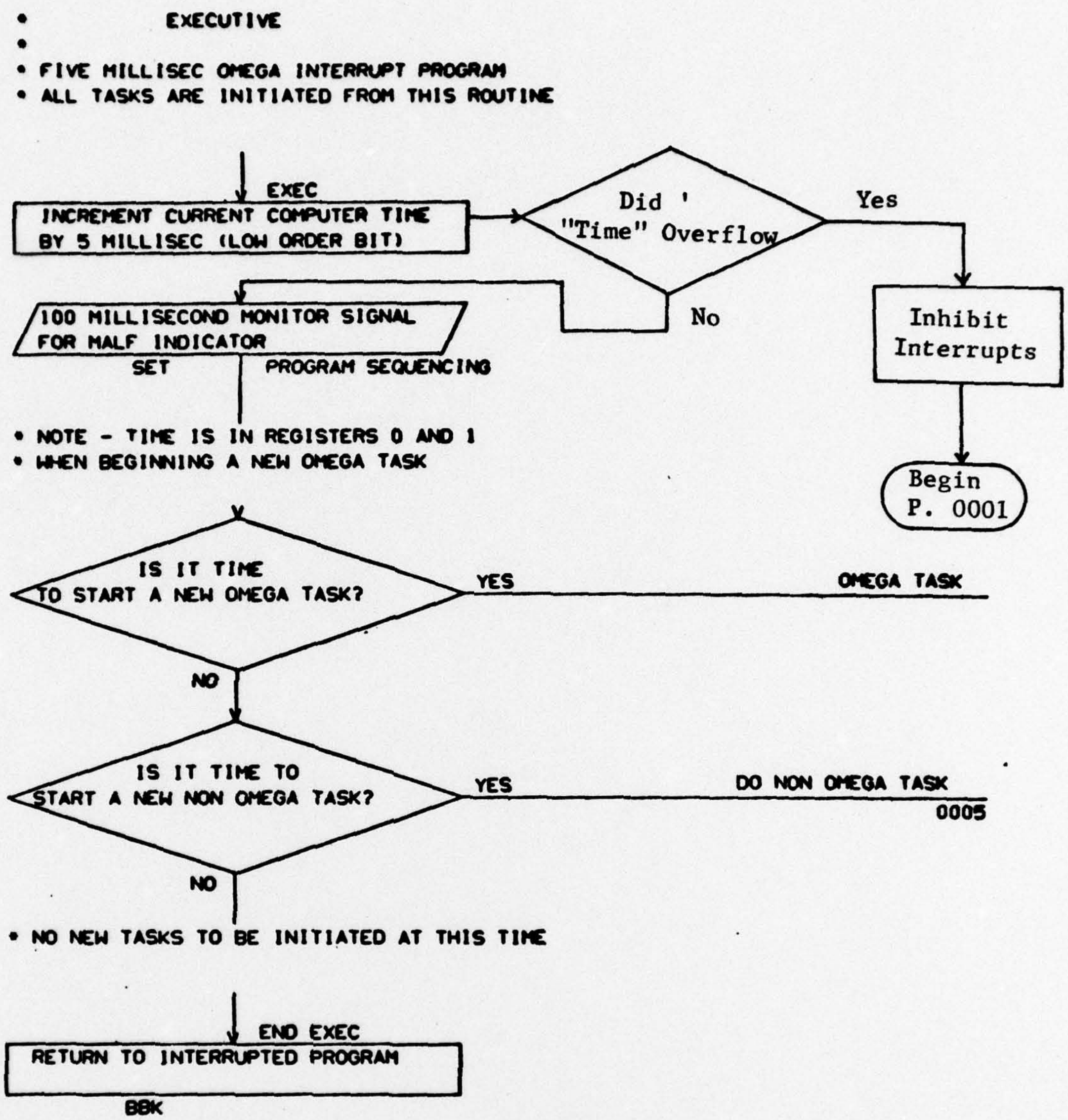
ADD VALUE OF COMPUTER TIME
EXISTING AT TURN OFF TO GMT
OF PREVIOUS TURN ON TO
GET TIME OF THIS TURN ON
(WHICH THE OPERATOR CAN REJECT)
CLEAR LOWER MEMORY FROM ZERO UP
TO AND INCLUDING TIME LOCATION

CLEAR UPPER MEMORY FROM THE
BOTTOM OF THE R15 PUSHDOWN STACK
UPTO PANEL DATA LOCATION

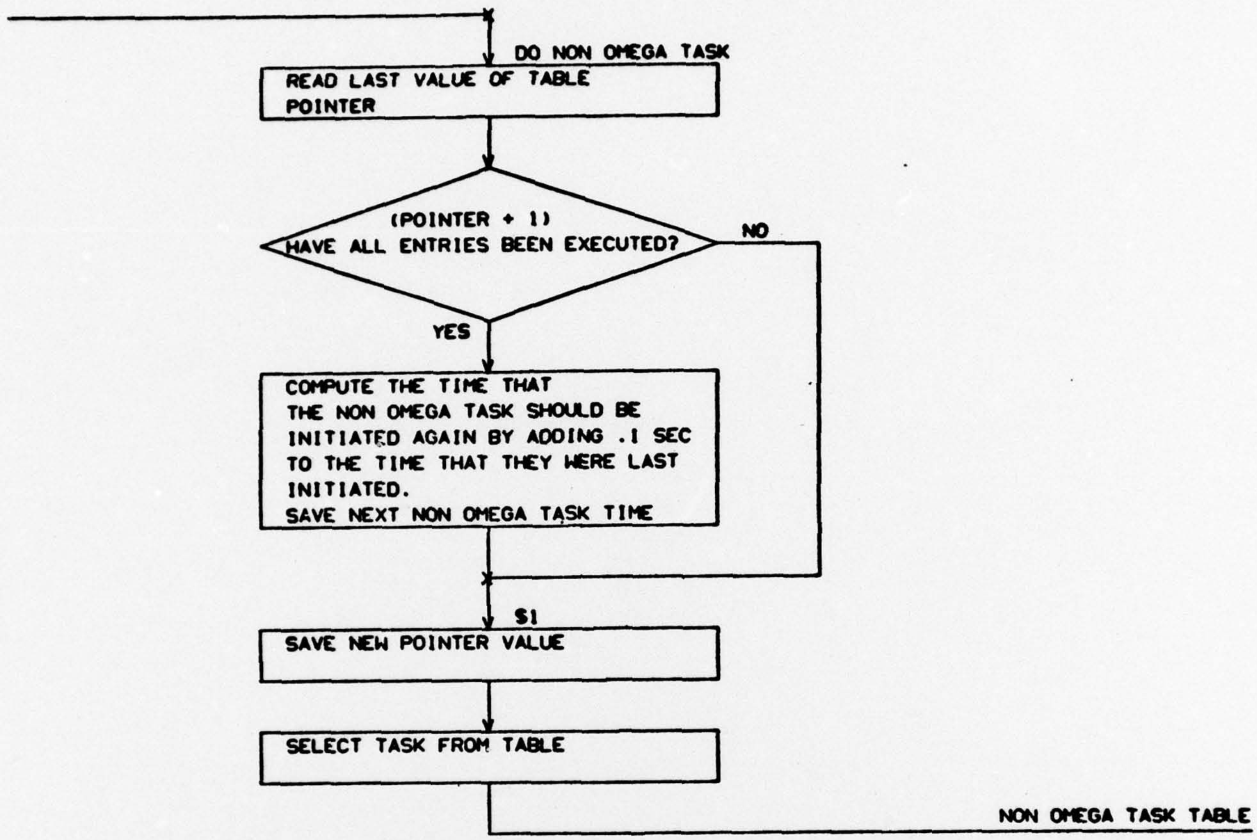

TURN ON THE INSERT, DISPLAY, NORMAL, ENTER T AND CLEAR LAMPS ON THE C/I PANEL
SET PANEL SWITCH MASK SO THAT DEPRESSING THE ILLUMINATED BUTTONS IS VALID
Display Program ID on C-I Panel. Delay for oscillator
SET 1ST OMEGA TASK = PREFLIGHT TO OCURR IN 50 MSEC
OMEGA
TEST1
INITIALIZE PRECISION FREQUENCY GENERATOR IN RECEIVER
SET
RESET PFG COUNTDOWN
ALLOW DIRECT MEMORY ACCESS I/O TO OCCUR (INHIBIT NOT TRUE)
LGC
DMA INHIBIT
ALLOW INTERRUPTS TO OCCUR IN THE 1070 (INHIBIT NOT TRUE)
LGC
INTERRUPT INHIBIT
SET COMPUTER AND SYSTEM OK BITS IN TEST SIGNAL OUTPUT WORD
100 MILLISECOND MONITOR SIGNAL FOR MALF INDICATOR
SET
PROGRAM SEQUENCING
• THE 5 MSEC OMEGA INTERRUPT WILL BE
• ENABLED AFTER ONE PASS THROUGH
• CHECKSUM. THIS ROUTINE WILL CYCLE
• THROUGH CHECKSUM REPEATEDLY (EXCEPT
• WHEN DOING SYNC CALCULATIONS) WHEN
• THERE ARE NO OTHER PROCESSING REQUIREMENTS
TEST SYNC
CHECKSUM PROGRAM AREA OF MEMORY INDICATE MALF IF NOT CORRECT
BSV
MP CKSUM
HAS A SYNC 10 SEC DATA COLLECTION PERIOD JUST ENDED?
NO
YES
• PERFORM SYNC COMPUTATIONS
• RETURN TO TEST SYNC IF NOT COMPLETE

• CONTINUE ON IF SYNC IS COMPLETE



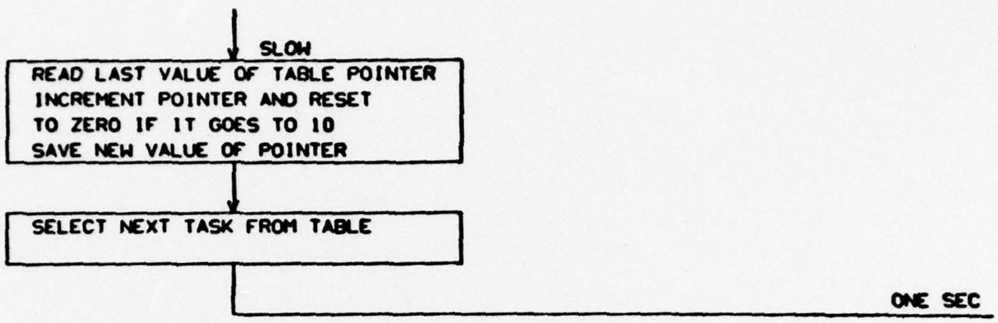

* EXECUTIVE
*
* FIVE MILLISEC OMEGA INTERRUPT PROGRAM
* ALL TASKS ARE INITIATED FROM THIS ROUTINE
EXEC
INCREMENT CURRENT COMPUTER TIME BY 5 MILLISEC (LOW ORDER BIT)
Did "Time" Overflow
Yes
No
Inhibit Interrupts
Begin P. 0001
100 MILLISECOND MONITOR SIGNAL FOR MALF INDICATOR
SET
PROGRAM SEQUENCING
* NOTE - TIME IS IN REGISTERS 0 AND 1
* WHEN BEGINNING A NEW OMEGA TASK
IS IT TIME TO START A NEW OMEGA TASK?
YES
OMEGA TASK
NO
IS IT TIME TO START A NEW NON OMEGA TASK?
YES
DO NON OMEGA TASK
0005
NO
* NO NEW TASKS TO BE INITIATED AT THIS TIME
END EXEC
RETURN TO INTERRUPTED PROGRAM
BBK

* NON OMEGA TASK
*
* THIS ROUTINE SELECTS THE NEXT NON OMEGA TASK
* FROM A TABLE. AFTER ALL TASKS HAVE BEEN
* EXECUTED IT UPDATES THE TIME OF NON OMEGA
* TASKS BY .1 SEC.



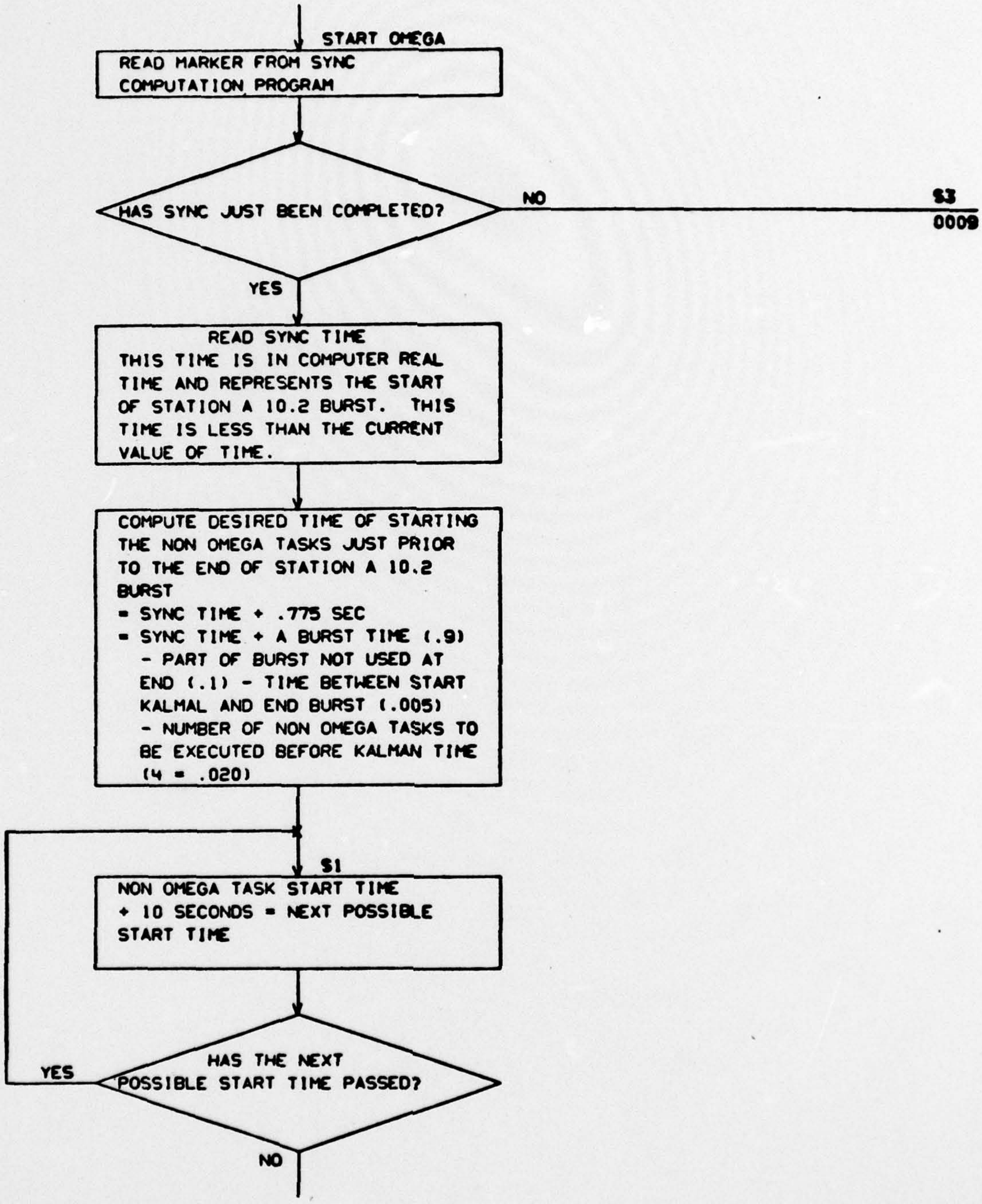
* THE TABLE CONSISTS OF 12 ENTRIES	OMEGA DEBUG
*	VELOCITY PROCESSING
* THE SECOND ENTRY IS THE FIRST	OMEGA DEBUG
* ROUTINE EXECUTED AND THE FIRST	RIJ UPDATE
* ENTRY IS THE LAST ROUTINE	OMEGA DEBUG
* EXECUTED IN A GIVEN .1 SEC	OMEGA DEBUG
* PERIOD. AFTER THE LAST ROUTINE	PANEL MAIN
* IS EXECUTED THERE IS A WAIT	OMEGA DEBUG
* (USUALLY .04 SEC) BEFORE THE	AD IO TEST
* FIRST ROUTINE IS EXECUTED AGAIN.	OMEGA DEBUG
* ONE ROUTINE IS EXECUTED EACH 5	SLOW
* MILLISECONDS	START OMEGA

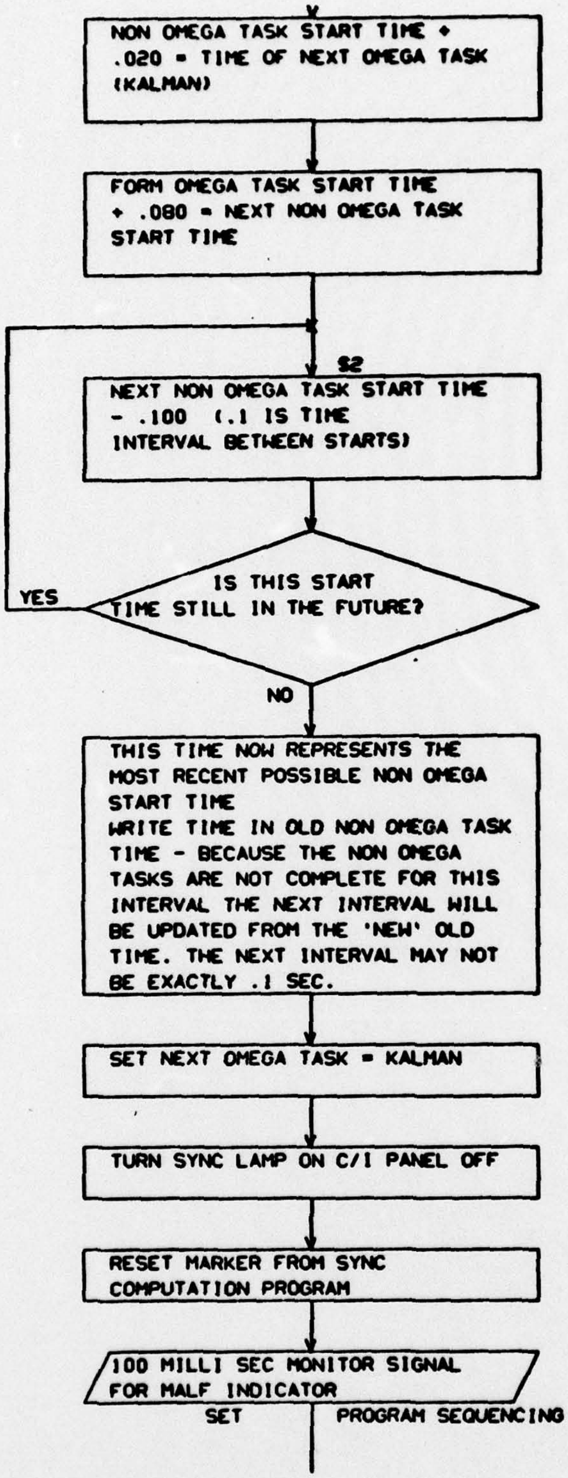
- ONE SECOND
-
- THIS ROUTINE IS A NON OMEGA TASK THAT
- WILL CYCLE THROUGH A TABLE OF TEN ENTRIES
- EXECUTING ONE TASK EACH TIME (THEREBY
- YIELDING ROUTINES EXECUTED ONCE A SECOND)

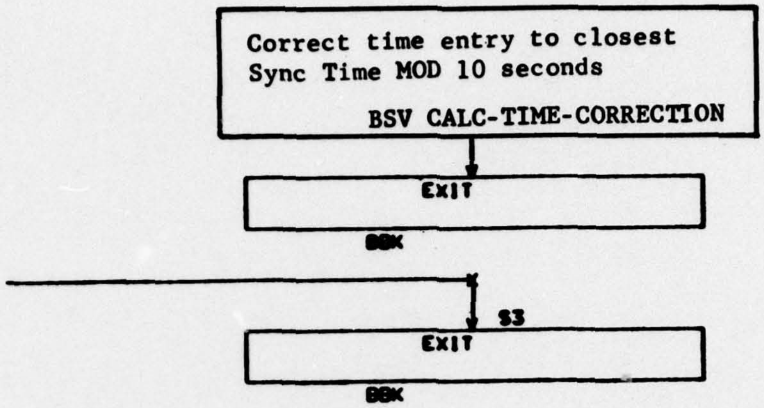


• THE TABLE CONSISTS OF 10 ENTRIES	RZ UPDATE	2ND
• WHERE THE Unused Entries	RZ TIME UPDATE	3RD
• RETURN TO THE INTERRUPTED PROGRAM	GP TEST	5TH
• AS THERE ARE ONLY 5 1-SEG ROUTINES	OSC Test	6th
	OPER PC	8th

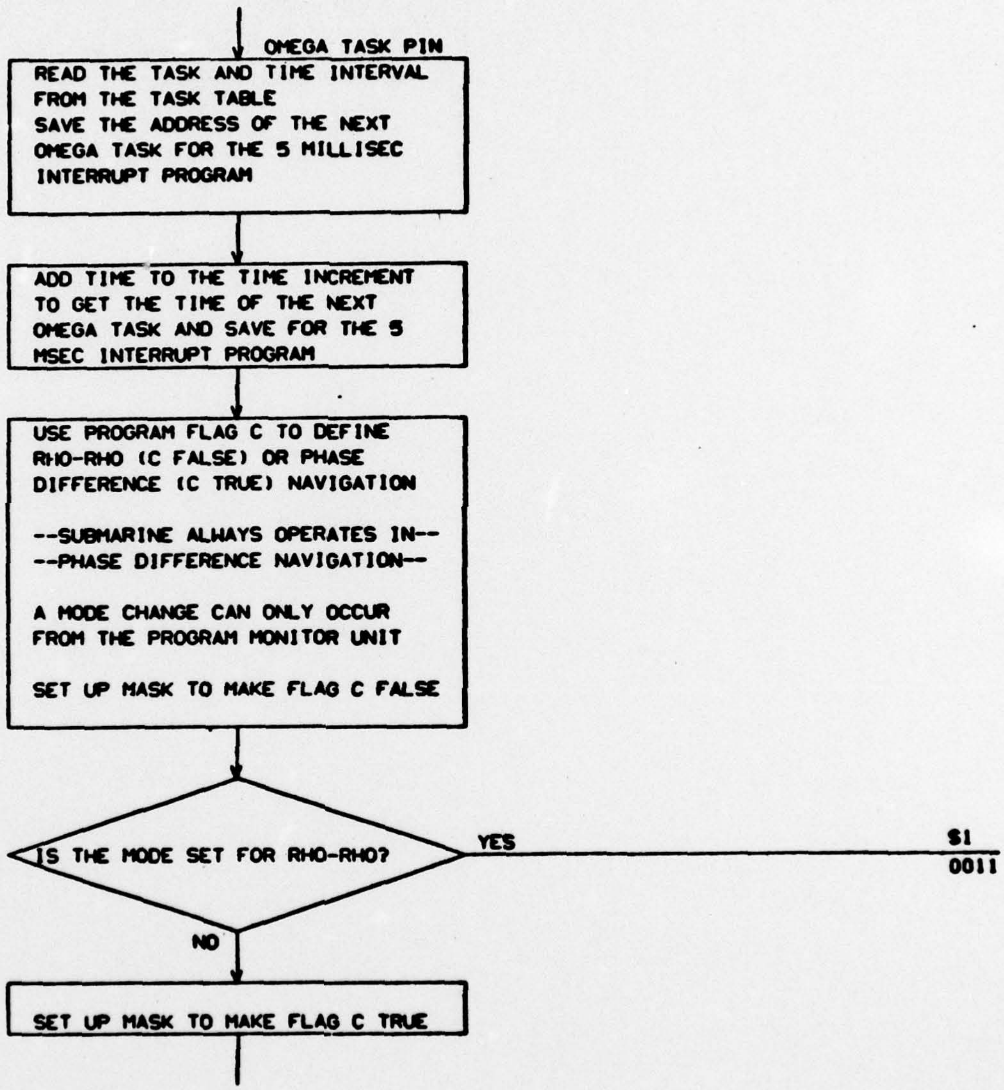
- START OMEGA NAVIGATION
-
- THIS ROUTINE TRANSLATES THE TIME DETERMINED
- BY THE SYNC PROCESS INTO THE TIME THAT THE
- OMEGA NAVIGATION TASKS SHOULD BE PERFORMED




NON OMEGA TASK START TIME + .020 = TIME OF NEXT OMEGA TASK (KALMAN)
FORM OMEGA TASK START TIME + .080 = NEXT NON OMEGA TASK START TIME
S2
NEXT NON OMEGA TASK START TIME - .100 (.1 IS TIME INTERVAL BETWEEN STARTS)
IS THIS START TIME STILL IN THE FUTURE?
YES
NO
THIS TIME NOW REPRESENTS THE MOST RECENT POSSIBLE NON OMEGA START TIME
WRITE TIME IN OLD NON OMEGA TASK TIME - BECAUSE THE NON OMEGA TASKS ARE NOT COMPLETE FOR THIS INTERVAL THE NEXT INTERVAL WILL BE UPDATED FROM THE 'NEW' OLD TIME. THE NEXT INTERVAL MAY NOT BE EXACTLY .1 SEC.
SET NEXT OMEGA TASK = KALMAN
TURN SYNC LAMP ON C/I PANEL OFF
RESET MARKER FROM SYNC COMPUTATION PROGRAM
100 MILLI SEC MONITOR SIGNAL FOR HALF INDICATOR
SET
PROGRAM SEQUENCING


Correct time entry to closest
Sync Time MOD 10 seconds
BSV CALC-TIME-CORRECTION
EXIT
BBX
S3
EXIT
BBX

* OMEGA TASK
*
* THIS PIN ROUTINE SELECTS THE NEXT OMEGA
* TASK FROM A TABLE AND COMPUTES THE TIME
* OF THE NEXT OMEGA TASK FROM THE TIME
* INCREMENT IN THE TABLE
* THE ARGUMENTS CONSIST OF A POINTER TO
* THE TABLE AND THE CURRENT REAL TIME

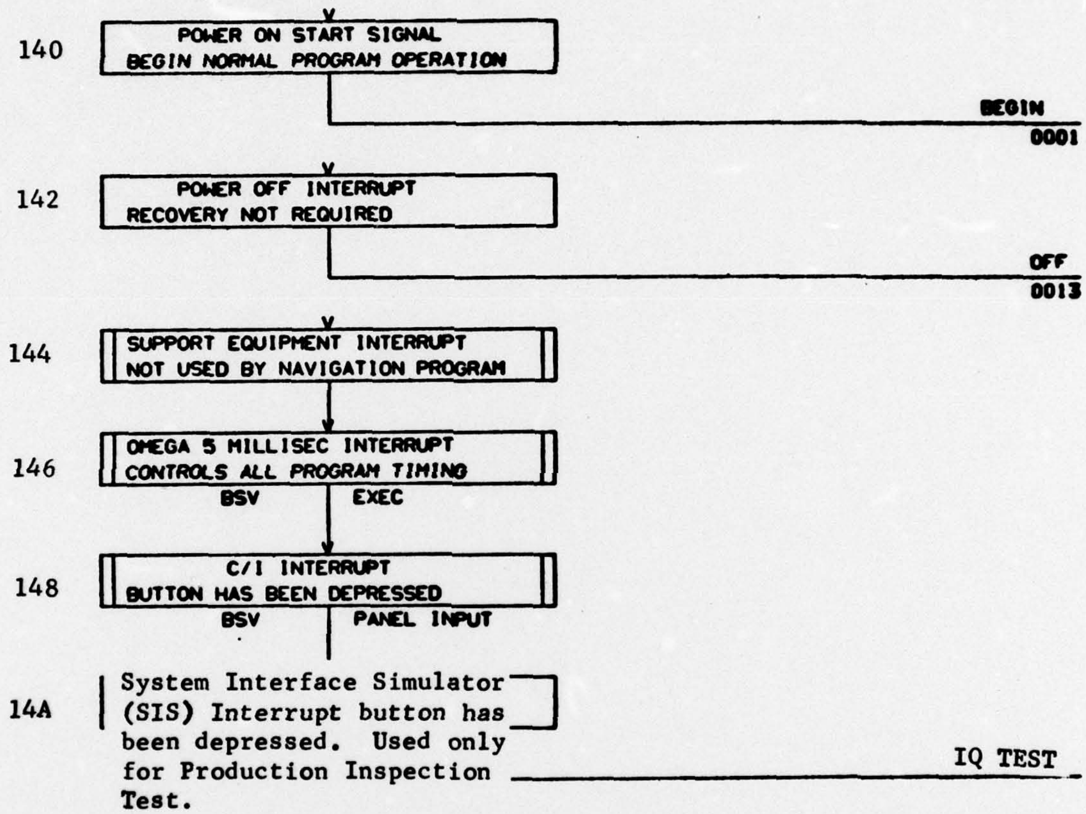


S1

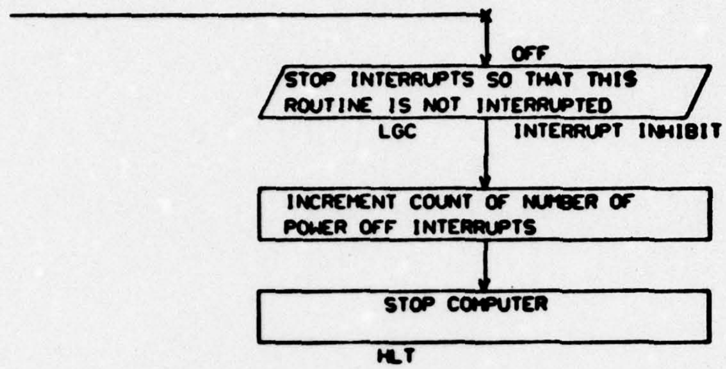
SET SELECTED STATE OF FLAG C IN THE APPROPIATE BIT IN THE INDICATOR WORD OF THE R14 SUBROUTINE RETURN PUSHDOWN STACK AFTER EXITING THIS PIN ROUTINE WITH A BRANCH BACK FLAG C WILL HAVE THE PROPER STATE

BBK

•   INTERRUPTS
•
• THE FOLLOWING INSTRUCTIONS ARE LOCATED IN THE WORDS THAT ARE HARDWIRED
• IN THE COMPUTER FOR INTERRUPTS AND START SIGNALS. THE START SIGNALS
• CAN BE GIVEN BY EITHER POWER ON OR THE MEMORY AUGMENTER. THE FIRST
• INSTRUCTION AFTER A START SIGNAL MUST LOAD THE NEXT INSTRUCTION
• REGISTER. WHEN INTERRUPTS OCCUR THE NEXT INSTRUCTION REGISTER WILL
• CONTAIN THE LOCATION OF THE INSTRUCTION THAT WOULD HAVE BEEN EXECUTED
• NEXT IF THE INTERRUPT HAD NOT OCCURRED. THE FIRST INSTRUCTION AFTER
• THE INTERRUPT MAY PRESERVE THIS LOCATION IF REQUIRED.
•



```
*         POWER OFF
*
* POWER OFF INTERRUTS FORCE ENTRY TO THIS ROUTINE
*
```



3.3 COMPUTER SUBPROGRAM ENVIRONMENT

3.3.1 Executive Tables

a) Non-OMEGA Routines: This table contains pointers to each of the Non-OMEGA tasks. It is defined in detail in the flow charts and in the listing. The official name of this table is NON-OMEGA TASK TABLE.

b) One-Second Routines: This table contains pointers to each of the one-second routines. It is defined in detail in the flow charts and in the listing. The official name of this table is ONE SEC.

c) OMEGA Tasks: This table contains an entry for every OMEGA task. Each entry has two items. The first is the pointer to the task and the second is the delta-t-OMEGA that should be added to the time argument of the OMEGA Task Pin. This table is defined in the listing. The official name of this table is OMEGA TASK TABLE.

3.3.2 Executive Temporary Storage

All temporary storage used by the Executive program is in the R15 pushdown stack.

3.3.3 Input/Output

Not applicable.

3.3.4 Required System Library Subroutine

Subroutine	Flow Diagram Reference	Subprogram Design Document (Volume Number)
MP CHECKSUM	P2/TEST SYNC P3/MAIN PROGRAM	XI